

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-215179

(43)Date of publication of application : 04.08.2000

(51)Int.Cl. G06F 15/16  
G06F 13/00

(21)Application number : 11-013935

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 22.01.1999

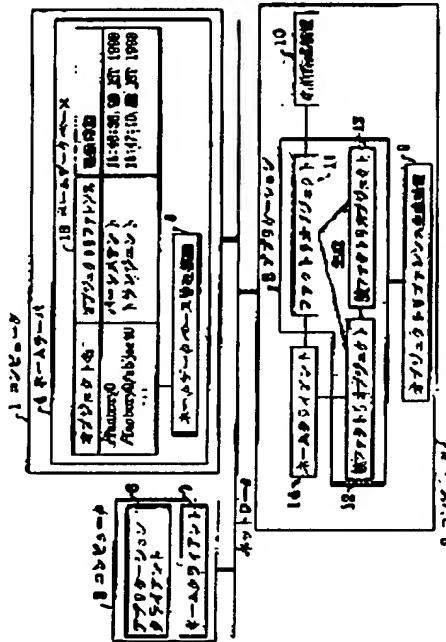
(72)Inventor : OTANI HARUYUKI

## (54) MANAGEMENT SYSTEM OF OBJECT NAME

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a management system of object names which can decrease the network traffic when the registered information on the factory objects and objects to be made factory which are registered on a name data base of a name server via a network.

**SOLUTION:** An object reference generating device 9 generates the object references to identify both factory and objects to be made factory respectively, and a name production device 10 produces the name of the object reference and the name of an object to be made factory that is connected to the name of the object reference. The object references and names of the factory and objects to be made factory are registered on a name data base 16. A name data base management device deletes the identification information and the name of the factory object based on the name of the factory object and also deletes the identification information and the name of the object to be made factory based on the name of the object to be made factory that is connected to the name of the factory object.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

2006年8月8日 16時10分

ASAMURA 81-332461239

Searching PAJ

NO. 9690 P. 64/135

2/2 ページ

[decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (JP)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2000-215179

(P2000-215179A)

(43) 公開日 平成12年8月4日(2000.8.4)

(51) Int Cl' 異別記号  
G 0 6 F 15/16 6 2 0  
13/00 3 5 3

F I テ-マ-ト<sup>®</sup>(参考)  
G O 6 F 15/16 6 2 0 T 5 B 0 4 5  
13/00 3 5 3 C 5 B 0 8 9

審査請求・審査請求の裏面 Q1 (合 15 頁)

(21) 出願番号 特願平11-13935

(22) 出願日 平成11年1月22日(1999.1.22)

(71) 出國人 000006013

三重電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 実明者 大谷 治之

東京都千代田区丸

## 麥電機株式会社内

100102439

金程士 富用 金程 (外2名)

卷二

६३१

5B089 GA11

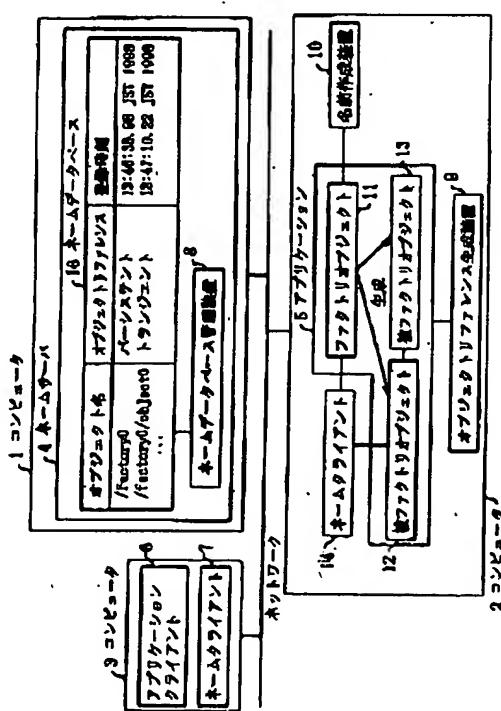
KC51 KD02

(54) [発明の名称] オブジェクト名前管理方式

(57) 【要約】

**【課題】** ネットワークを通じてネームサーバのネームデータベースに登録されているファクトリオプロジェクトと被ファクトリオプロジェクトの登録情報を削除する際、ネットワークトラフィックの削減ができるオブジェクト名前管理方式を実現する。

【解決手段】 ファクトリオブジェクトと被ファクトリオブジェクトのそれぞれを識別するオブジェクトリファレンスをオブジェクトリファレンス生成装置9で生成し、ファクトリオブジェクトの名前とこのファクトリオブジェクトの名前が連結された被ファクトリオブジェクトの名前を名前作成装置10で作成し、これらのファクトリオブジェクトと被ファクトリオブジェクトのオブジェクトリファレンスと名前をネームデータベース16に登録し、また、ネームデータベース管理装置はファクトリオブジェクトの名前に基づきファクトリオブジェクトの識別情報と名前を削除すると共に、ファクトリオブジェクトの名前が連結された被ファクトリオブジェクトの名前に基づき被ファクトリオブジェクトの識別情報と名前を削除する。



## 【特許請求の範囲】

【請求項1】 以下の要素を備えたオブジェクト名前管理方式。

(a) 以下の要素を有するサーバ；

(a 1) 起動時に生成されるファクトリオブジェクトと上記ファクトリオブジェクトにより生成される被ファクトリオブジェクトのそれぞれの識別情報と名前の登録を要求し、また登録された上記識別情報と上記名前の削除の要求をするアプリケーション；

(a 2) 上記ファクトリオブジェクトが生成された後に上記ファクトリオブジェクトの上記識別情報を生成し、上記被ファクトリオブジェクトが生成された後に上記被ファクトリオブジェクトの上記識別情報を生成するオブジェクトリファレンス生成手段；

(a 3) 上記ファクトリオブジェクトの識別情報が生成された後に上記ファクトリオブジェクトの名前を作成し、上記被ファクトリオブジェクトの識別情報が生成された後に上記被ファクトリオブジェクトの名前を作成し、この名前に上記ファクトリオブジェクトの名前を連結する名前作成手段；

(b) 以下の要素を有するネームサーバ；

(b 1) 上記ファクトリオブジェクトと上記被ファクトリオブジェクトのそれぞれの上記識別情報と上記名前が登録されるネームデータベース；

(b 2) 上記アプリケーションにより登録を要求された上記ファクトリオブジェクトと上記被ファクトリオブジェクトのそれぞれの上記識別情報と上記名前を受信して上記ネームデータベースに登録し、また、上記ファクトリオブジェクトの名前に基づき上記ファクトリオブジェクトの識別情報と名前を削除すると共に、上記ファクトリオブジェクトの名前が連結された上記被ファクトリオブジェクトの名前に基づき上記被ファクトリオブジェクトの識別情報と名前を削除するネームデータベース管理手段；

【請求項2】 上記オブジェクトリファレンス生成手段は、上記被ファクトリオブジェクトの識別情報に上記被ファクトリオブジェクトの生成された生成時刻を付加し、上記ネームデータベース管理手段は、上記生成時刻に基づいて上記ネームデータベースに登録されている上記被ファクトリオブジェクトの有無を確認し、確認された被ファクトリオブジェクトの上記識別情報と上記名前を削除することを特徴とする請求項1記載のオブジェクト名前管理方式。

【請求項3】 上記ファクトリオブジェクトと上記被ファクトリオブジェクトのそれぞれの上記識別情報と上記名前が保存されるネームデータベースキャッシュと、新たな上記更新情報と上記名前を上記ネームデータベースキャッシュに既に保存されている上記ファクトリオブジェクトの識別情報と名前に上書きし、上記ネームデータベースキャッシュに既に保存されている上記ファクト

リオブジェクトに対応する上記被ファクトリオブジェクトの識別情報と名前を削除するネームクライアントとを備え、

上記ネームデータベース管理手段は、上記ネームデータベースに登録された上記ファクトリオブジェクトの識別情報と名前の更新に基づいた新たな識別情報と名前を上記ネームクライアントに送信することを特徴とする請求項1記載のオブジェクト名前管理方式。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 この発明は、分散オブジェクトシステムにおけるネーミングサービスに関するものである。

## 【0002】

【従来の技術】 図11は例えば、「CORBA services Marc h1995 Naming Service Specification」に示された分散オブジェクトシステムにおける従来のネーミングサービスの構成図である。分散オブジェクトシステムは、2つのコンピュータ102、103がネットワークによって相互に接続されたコンピュータシステムで、コンピュータ102内のアプリケーション105、アプリケーション105内のオブジェクト111、112、113及びオブジェクト111、112、113に対してオブジェクトリファレンス生成装置109と、コンピュータ103内のアプリケーションクライアント106と、により構成される。

【0003】 アプリケーションクライアント106はオブジェクトリファレンス生成装置109によって生成されたオブジェクトリファレンスを用いアプリケーション105内のオブジェクト111、112、113を特定し、特定されたオブジェクトとメッセージを送受信することにより、アプリケーションに固有の処理を実行させる。

【0004】 分散オブジェクトシステムにおけるネーミングサービスは、上記の分散オブジェクトシステムに加えて、第1のコンピュータ101がネットワークを通じてコンピュータ102及びコンピュータ103と相互に接続されたもので、コンピュータ101内で動作するネームサーバ104と、コンピュータ102内で動作するネームクライアント107及びコンピュータ103内で動作するネームクライアント114により構成される。

【0005】 ネームサーバ104は、オブジェクト111、112、113に対して割り当たられるオブジェクト名とオブジェクトリファレンスの組（以下、組と記載する）が登録されるネームデータベース110と、ネームデータベース110内の組を実際に登録したり、削除したり、検索したりする処理を行うネームデータベース管理装置108から成る。組の登録、削除の要求はアプリケーション105が行ない、組の検索の要求はアプリケーションクライアント106が行う。これらの要求

は、それぞれネームクライアント114、ネームクライアント107を通じてネームサーバ104に送られる。

【0006】図11では、ネームデータベース110は分散配置されたデータベースとして図示していないが、本来は、ネームサーバ104を複数設け、分散データベースとして構築することが可能である。

【0007】オブジェクトリファレンス生成装置109はオブジェクト111、112、113を全システム内で一意に識別するオブジェクトリファレンスを生成する。オブジェクトリファレンスの代表的な例はOMG CORBAのIOR(Interoperable Object Reference)が挙げられる。アプリケーションクライアント106はこのオブジェクトリファレンスを獲得することで、オブジェクト111、112、113にアクセスし、オブジェクト111、112、113とのメッセージの送受信が可能になる。

【0008】次に、図12を参照して動作について説明する。図12は、図11の従来のネーミングサービスの主要構成部の処理動作を時系列で表した動作説明図である。ここで、縦線は時間を表している。まず、アプリケーション105を起動すると、アプリケーション105の初期化の時点で、あらかじめプログラミングされているオブジェクト111が生成される。オブジェクト111が生成されると、オブジェクト111を特定するオブジェクトリファレンスがオブジェクトリファレンス生成装置109によって生成され一意に割り当てられる。

【0009】図11におけるオブジェクトリファレンスは、例えば、オブジェクトが存在するホスト名、オブジェクトに対するメッセージを送受信するポート番号及びオブジェクトキー(オブジェクトのインターフェース名、プログラムによって与えられる文字列など)により構成される。

【0010】次に、オブジェクトリファレンスとプログラミングによってあらかじめ与えられたオブジェクト名との組が、ネームクライアント114を通じて、ネームサーバ104のネームデータベース110に登録される(ステップS101)。ネームサーバ104のネームデータベース管理装置108は、アプリケーション105からの要求により組の登録を行うが、この際、登録しようとする名前と同一名のオブジェクトリファレンスが既にネームデータベース110内に存在するか否かを調べる。同一名のオブジェクトリファレンスが存在した場合は、2つの対応策が有り得る。1つは、重複エラーとして、登録を依頼してきたネームクライアント114にレポートするという場合であり、もう1つはアプリケーション105が強制登録を望む場合に限り、既存の組を削除して新規の組を登録するという場合である。

【0011】次に、アプリケーションクライアント106は、ネームクライアント107を通じて、あらかじめプログラミングによって与えられた名前を、ネームサー-

バ104に送り、対応するオブジェクトリファレンスを獲得する(ステップS102)。ネームサーバ104のネームデータベース管理装置108は、ネームクライアント107からの検索要求が来ると、オブジェクト名によってネームデータベース110を検索し、該当するオブジェクトリファレンスをネームクライアント107に送る。さらに、ネームクライアント107は送られてきたオブジェクトリファレンスをアプリケーションクライアント106に送る。

【0012】アプリケーションクライアント106は、ネームクライアント107からオブジェクトリファレンスを獲得すると、オブジェクト111が存在するホスト名、ポート番号、オブジェクトキーを、オブジェクトリファレンスより抽出し、ネットワークを通じてオブジェクト111とメッセージの送受信を行い、アプリケーション105に処理を遂行させる(ステップS106)。

【0013】アプリケーション105の処理内容に依存するが、オブジェクト111に対して、他のオブジェクトを生成するように要求することがある(ステップS103)。図11では、アプリケーション105の起動時に生成された最初のオブジェクト111を含めて合計3つのオブジェクトが生成されている様子を示している。新たにオブジェクト112、113が生成されると、最初のオブジェクト111と同様に、オブジェクトリファレンス生成装置109によって新しいオブジェクトリファレンスが生成され、この生成されたオブジェクトリファレンスとあらかじめプログラミングによって与えられた名前とにより構成された組がネームデータベース110に登録される(ステップS104、S105)。

【0014】アプリケーション105の終了時(正常に終了する場合)には、アプリケーション105はネームクライアント114を通じて、ネームデータベース110から組を削除するよう、ネームサーバ104へ要求を行う。この要求を受け取ったネームサーバ104のネームデータベース管理装置108は、ネームクライアント114から受け取ったオブジェクト名によって、ネームデータベース110を検索し、該当するオブジェクト名とオブジェクトリファレンスとの組を削除する(ステップS1307、S1308)。

### 【0015】

【発明が解決しようとする課題】以上説明したように、従来のネーミングサービスシステムでは、例えばアプリケーション105の起動時に生成されるオブジェクト111とこの生成されたオブジェクト111によってさらに生成されるオブジェクト112、113とは同じオブジェクトとして扱われ、また、オブジェクトリファレンスもこれらの種別を区別せず生成されていた。このため、次のような問題点が生じた。

【0016】すなわち、ネームデータベース110内では、どのオブジェクトも種別を区別されないため、ネ-

ムデータベース110から組を削除する場合は、それぞれ個々の組に対して、削除の要求を出す必要があった。このため、ネームサーバ104がアプリケーション105と別のコンピュータで動作する場合、ネームデータベース110内の組の数に応じて削除の要求がネットワーク上を往来し、ネットワークトラフィックを増大するという問題点があった。

【0017】また、アプリケーション105が異常終了してしまうと、ネームデータベース110内に無効の組が残る。アプリケーション105を再起動した場合には、これらの組を全て削除し再度新しい組を登録する必要があるが、無効の組が多量に残存した場合は、これらの組を全て削除しなければならず、アプリケーション105の回復時間に多大な時間を費やすという問題点があった。

【0018】さらに、ネームデータベース110に登録されている組の一部又は全体がコピーされて保存されるキャッシュが別のコンピュータに保持された場合、ネームデータベース110とそのコピーであるキャッシュのデータ内容の一貫性を維持するために、ネームデータベース110に登録されている組の更新に際しては、更新情報を全てをキャッシュにも送信し、この更新情報をキャッシュに反映する必要があった。このように全ての更新情報をネットワークを通じてキャッシュに送信する必要があったため、ネットワークトラフィックが大きくなる問題点があった。

【0019】また、キャッシュは必ずしも最新の状態とは限らず、更新情報が反映されるまでか、あるいは、キャッシュ内のオブジェクト名とオブジェクトリファレンスを用いて実際に処理を遂行してみるまでは、そのデータが有効なものか無効なものかわからないという問題点があった。

【0020】さらに、従来のキャッシュとネームデータベース110とのデータの一貫性を維持する方法では、新しくオブジェクト名とオブジェクトリファレンスの組がネームデータベース110に登録された場合、ネームデータベース110内の新しいオブジェクト名とオブジェクトリファレンスを更新情報としてキャッシュに送っていたが、これによってネットワークトラフィックが大きくなり、また、更新情報をキャッシュに反映する処理にも時間がかかるという問題点があった。

【0021】この発明は、上記のような問題を解決するためになされたもので、ネットワークを通じてネームデータベースの登録情報を削除する際のネットワークトラフィックの削減ができ、また異常終了したアプリケーションを再起動する際の回復時間の短縮ができ、さらにネームデータベースとキャッシュに保管の組を効率的に一貫管理することのできるオブジェクト名前管理方式を提供することを目的とする。

【0022】

【課題を解決するための手段】第1の発明は、以下の要素を備えたものである。

(a) 以下の要素を有するサーバ：

(a 1) 起動時に生成されるファクトリオブジェクトと上記ファクトリオブジェクトにより生成される被ファクトリオブジェクトのそれぞれの識別情報と名前の登録を要求し、また登録された上記識別情報と上記名前の削除の要求をするアプリケーション：

(a 2) 上記ファクトリオブジェクトが生成された後に上記ファクトリオブジェクトの上記識別情報を生成し、上記被ファクトリオブジェクトが生成された後に上記被ファクトリオブジェクトの上記識別情報を生成するオブジェクトリファレンス生成手段：

(a 3) 上記ファクトリオブジェクトの識別情報が生成された後に上記ファクトリオブジェクトの名前を作成し、上記被ファクトリオブジェクトの識別情報が生成された後に上記被ファクトリオブジェクトの名前を作成し、この名前に上記ファクトリオブジェクトの名前を連結する名前作成手段：

(b) 以下の要素を有するネームサーバ：

(b 1) 上記ファクトリオブジェクトと上記被ファクトリオブジェクトのそれぞれの上記識別情報と上記名前が登録されるネームデータベース：

(b 2) 上記アプリケーションにより登録を要求された上記ファクトリオブジェクトと上記被ファクトリオブジェクトのそれぞれの上記識別情報と上記名前を受信して上記ネームデータベースに登録し、また、上記ファクトリオブジェクトの名前に基づき上記ファクトリオブジェクトの識別情報と名前を削除すると共に、上記ファクトリオブジェクトの名前が連結された上記被ファクトリオブジェクトの名前に基づき上記被ファクトリオブジェクトの識別情報と名前を削除するネームデータベース管理手段。

【0023】第2の発明は、上記被ファクトリオブジェクトの識別情報に上記被ファクトリオブジェクトの生成された生成時刻を付加するオブジェクトリファレンス生成手段と、上記生成時刻に基づいて上記ネームデータベースに登録されている上記被ファクトリオブジェクトの有無を確認し、確認された被ファクトリオブジェクトの上記識別情報と上記名前を削除するネームデータベース管理手段とを備えたものである。

【0024】第3の発明は、上記ファクトリオブジェクトと上記被ファクトリオブジェクトのそれぞれの上記識別情報と上記名前が保存されるネームデータベースキャッシュと、新たな上記更新情報と上記名前を上記ネームデータベースキャッシュに既に保存されている上記ファクトリオブジェクトの識別情報と名前に上書きし、上記ネームデータベースキャッシュに既に保存されている上記ファクトリオブジェクトに対応する上記被ファクトリオブジェクトの識別情報と名前を削除するネームクライ

アントとを備え、上記ネームデータベース管理手段は、上記ネームデータベースに登録された上記ファクトリオブジェクトの識別情報と名前の更新に基づいた新たな識別情報と名前を上記ネームクライアントに送信するものである。

#### 【0025】

【発明の実施の形態】実施の形態1. 図1は、実施の形態1のオブジェクト名前管理方式（ネーミングサービス）の構成図である。このネーミングサービスは、コンピュータ1、2、3が相互にネットワークで接続されたシステムにおいて、コンピュータ1内のネームサーバ4と、コンピュータ2内のアプリケーション5、オブジェクトリファレンス生成装置9、名前作成装置10及びネームクライアント14と、コンピュータ3内のアプリケーションクライアント6及びネームクライアント7により構成される。

【0026】オブジェクトには、コンピュータシステム全体内で個々のオブジェクトを識別するためのオブジェクトリファレンスと呼ばれる識別子が付いている。オブジェクトリファレンスの代表的な例としてはOMG CORBAのIOR（Interoperable Object Reference）がある。IORには、このオブジェクトが動作する、ホスト名とメッセージを受信するためのTCP/IPのポート番号とこのホスト及びポート内でオブジェクトを一意に識別するためのオブジェクトキー（インターフェース名、インターフェース名+サーバ名又はアプリケーション5によって付けられた固定の文字列）により構成されており、これらはコンピュータシステム内のオブジェクトを一意に識別する役割を持つ。

【0027】本発明によるオブジェクトリファレンス生成装置9は従来の方法と同様に生成されたオブジェクトに対してオブジェクトリファレンスを一意に割り当てる機能を持つが、従来の方法とは異なりオブジェクトをファクトリオブジェクト11と被ファクトリオブジェクト12、13に種別し、それぞれに対して異なる種類のオブジェクトリファレンスを生成する。

【0028】ファクトリオブジェクト11は、他のオブジェクトを生成する（あるいは生成と消滅の両方を行うこともある）オブジェクトである。本発明での厳密な定義は、アプリケーション5が起動し、アプリケーションクライアント6からの処理要求を受け付けることが可能になる前に、すなわちアプリケーション5の起動が完了する前に、あらかじめ行われたプログラミングにより自動的に生成されるオブジェクトを指す。また、本発明では、その後のアプリケーションクライアント6の要求に従って、ファクトリオブジェクト11によって生成されたオブジェクトを被ファクトリオブジェクト12、13と呼ぶ。

【0029】図1に示すように、ファクトリオブジェク

ト11と被ファクトリオブジェクト12、13はアプリケーション5内に存在する。アプリケーションクライアント6はこれらのオブジェクトに対してメッセージを送受信することで、アプリケーションに処理を遂行させる。

【0030】ネームクライアント7、14とネームサーバ4はネームサービスを提供する。ネームサーバ4は、ネームデータベース管理装置8とネームデータベース16により構成される。ネームデータベース管理装置8は、オブジェクトの名前（オブジェクト名）と、オブジェクトリファレンスと、オブジェクト名とオブジェクトリファレンスを受け取った時刻である登録時刻と、から構成される組（以下、組と記載する）をネームデータベース16に登録したり、あるいは、登録された組を削除するものである。ネームデータベース16はこれらの組の記憶部である。

【0031】ネームクライアント14は、アプリケーション5からの登録要求に従ってオブジェクトリファレンスとアプリケーション5の指定に基づいて作成されたオブジェクト名とをネームサーバ4のネームデータベース管理装置8に渡す。ネームデータベース管理装置8は、渡されたオブジェクト名とオブジェクトリファレンスを関連づけ、オブジェクト名とオブジェクトリファレンスにオブジェクト名とオブジェクトリファレンスを受け取った時刻である登録時刻を加えた組を、ネームサーバ4内のネームデータベース16に登録する。

【0032】アプリケーション5は、オブジェクト名を指定することで、ネームデータベース管理装置108にオブジェクト名とオブジェクトリファレンスの関連づけを断ち、ネームサーバ4が管理するネームデータベース16から組を削除させる要求をネームクライアント14に渡す。ネームクライアント14は、渡された要求をネームデータベース管理装置8に送信する。ネームデータベース管理装置8は、送信された要求に基づいてネームデータベース16から対応する組を削除する。

【0033】また、アプリケーションクライアント6は、オブジェクト名を指定することで、それに関連づけられたオブジェクトリファレンスをネームサーバ4が管理するネームデータベース16から獲得するための要求をネームクライアント7に渡す。ネームクライアント7は、渡された要求をネームデータベース管理装置8に送信する。ネームデータベース管理装置8は、送信された要求に基づいてネームデータベース16から得たオブジェクトリファレンスを獲得し、この獲得したオブジェクトリファレンスをアプリケーションクライアント6に渡す。

【0034】次に動作について詳細に説明する。まず、アプリケーション5の起動時にファクトリオブジェクト11が生成される。例えば、アプリケーション5の実行プロセスをコマンドラインによって起動し、起動が完了

するまでの間に、ファクトリオブジェクト11が生成される。その後、被ファクトリオブジェクト12、13は、アプリケーションクライアント6の要求によってファクトリオブジェクト11で随時生成される。例えば、アプリケーション5の起動完了後、アプリケーションクライアント6は、ファクトリオブジェクト11のオブジェクトリファレンスをネームサーバ4から入手する。次にアプリケーションクライアント6は、ファクトリオブジェクト11に対して、新たなオブジェクトを生成するようにメッセージを送信する。これによって、被ファクトリオブジェクト12、13が生成される。

【0035】アプリケーション5が起動され、ファクトリオブジェクト11が生成されると、次にファクトリオブジェクト11に対応するオブジェクトリファレンスが生成され、さらにファクトリオブジェクト11のオブジェクト名が作成され、これらオブジェクトリファレンスとオブジェクト名がネームサーバ4のネームデータベース16に登録される。

【0036】ファクトリオブジェクト11が生成されてからファクトリオブジェクト11のオブジェクトリファレンスとオブジェクト名がネームサーバ4のネームデータベース16に登録されるまでの手順を以下に説明する。まず、アプリケーション5の起動により、ファクトリオブジェクト11が生成される。この際、オブジェクトリファレンス生成装置9が呼び出され、ファクトリオブジェクト11に割り当てるオブジェクトリファレンスが生成される。

【0037】図2はオブジェクトリファレンスの構成を示す図である。オブジェクトリファレンス生成装置9は、ファクトリオブジェクト11に対して、図2に示すようなパーシステントオブジェクトリファレンス21を生成する。パーシステントオブジェクトリファレンス21は、例えば、ホスト名、TCP/IPポート番号、オブジェクトを示す変更されない文字列（インターフェース名）により構成されるオブジェクトリファレンスである。ここで、オブジェクトリファレンスがパーシステントであるとは、アプリケーション5を再度起動した場合も同じオブジェクトに対しては同じオブジェクトリファレンスが生成されることを意味する。

【0038】オブジェクトリファレンス生成装置9は、オブジェクトリファレンスを割り当てる際、対象となるオブジェクトがファクトリオブジェクト11であるか否かを、アプリケーション5の起動完了時刻と、オブジェクトの生成時刻を比べることにより判断することができる。

【0039】図3は、オブジェクトリファレンス生成装置9において、オブジェクトリファレンスを生成する時の動作手順を示すフローチャートである。このフローチャートに従って、ファクトリオブジェクト11がオブジェクトリファレンスを生成する動作について説明する。

まず、オブジェクトリファレンス生成装置9は、アプリケーション5の起動完了時刻を入手し（ステップS1）、次にオブジェクトの生成時刻を入手する（ステップS2）。これらの時刻を比較して（ステップS3）、オブジェクトの生成時刻がアプリケーション5の起動完了時刻よりも前の場合、オブジェクトをファクトリオブジェクト11であると判断し、パーシステントオブジェクトリファレンス21を生成する（ステップS4）。

【0040】一方、アプリケーション5の起動完了時刻とオブジェクトの生成時刻を比べて、オブジェクトの生成時刻が後の場合、オブジェクトは被ファクトリオブジェクト12、13であると判断し、トランジエントオブジェクトリファレンス22を生成する（ステップS5）。被ファクトリオブジェクト12のオブジェクトリファレンスの生成については後述する。

【0041】図4は、名前作成装置10において、オブジェクトに与えるオブジェクト名を作成する時の動作手順を示すフローチャートである。このフローチャートに従ってファクトリオブジェクト11がオブジェクト名を作成する動作について説明する。まず、オブジェクトがファクトリオブジェクト11であるか判断して（ステップS11）、オブジェクトがファクトリオブジェクト11である時には、アプリケーション5によって付けられたオブジェクト名をそのまま適用することによって（ステップS12）、ファクトリオブジェクト11のオブジェクト名を作成する。図1では、ファクトリオブジェクト11に対してアプリケーション5によって指定された/factory0というオブジェクト名を付けたことを表している。

【0042】アプリケーション5は、以上のように生成されたファクトリオブジェクト11のオブジェクト名と、作成されたファクトリオブジェクト11のオブジェクト名と、をネームクライアント14に渡し、ネームサーバ4のネームデータベース16に登録するよう要求する。ネームクライアント14は、このオブジェクト名とオブジェクトリファレンスをネームデータベース管理装置8に送信し、ネームデータベース管理装置8がこれらをネームサーバ4のネームデータベース16に登録する。

【0043】アプリケーションクライアント6は、オブジェクト名/factory0を指定し、ネームサーバ4からファクトリオブジェクト11のオブジェクトリファレンスを入手する。その後、入手したオブジェクトリファレンスをもとにファクトリオブジェクト11に接続し、ファクトリオブジェクト11に対して、別のオブジェクトを生成するように要求を行なう。図1は、ファクトリオブジェクト11が2つの被ファクトリオブジェクト12、13を生成した様子を表している。

【0044】被ファクトリオブジェクト12、13が生成されると、ファクトリオブジェクト11と同様に、オ

プロジェクトリファレンス生成装置9により生成された被ファクトリオブジェクト12、13に対して一意に割り当てるオブジェクトリファレンスが生成される。オブジェクトリファレンス生成装置9において、被ファクトリオブジェクト12、13のオブジェクトリファレンスを生成する時の動作は、ファクトリオブジェクト11がオブジェクトリファレンスを生成する時の動作と同様であり、図3のフローチャートの通りである。この動作について説明する。

【0045】まず、オブジェクトリファレンス生成装置9は、アプリケーション5の起動完了時刻を入手し（ステップS1）、さらにオブジェクトの生成時刻を入手する（ステップS2）。次に、プログラムの起動完了時刻とオブジェクト生成時刻を比べる（ステップS3）。この場合、オブジェクト生成時刻は、プログラムの起動完了時刻よりも後である。従って、オブジェクトを被ファクトリオブジェクト12、13であると判断する。次に、オブジェクトリファレンス生成装置9は、オブジェクトが被ファクトリオブジェクト12、13であるという事実に基づいて、オブジェクトリファレンスを生成する（ステップS5）。

【0046】被ファクトリオブジェクト12、13に対するオブジェクトリファレンスは、図2に示す構成のトランジエントオブジェクトリファレンス22であり、先に示したファクトリオブジェクト11に対するパーシステントオブジェクトリファレンス21とは異なり、オブジェクトが生成された生成時刻を含む。従って、被ファクトリオブジェクト12、13のオブジェクトリファレンスはトランジエント（遷移的）であり、オブジェクトが生成されるたびに異なるオブジェクトリファレンスが生成される。

【0047】次に、名前作成装置10を利用してオブジェクト12、13に与えるオブジェクト名が作成される。名前作成装置10において、被ファクトリオブジェクト12、13のオブジェクト名を作成する時の動作は、ファクトリオブジェクト11がオブジェクト名を作成する時の動作と同様であり、図4のフローチャートの通りである。この動作について説明する。

【0048】名前作成装置10は、オブジェクトがファクトリオブジェクト11であるか被ファクトリオブジェクト12、13であるかを判断する（ステップS11）。このファクトリオブジェクト11であるか被ファクトリオブジェクト12、13であるかの判断は、オブジェクトリファレンス生成装置9において、ファクトリオブジェクト11がオブジェクトリファレンスを生成する時の動作と同様である。すなわち、図3のフローチャートのステップS1～S3と同じ動作手順によってファクトリオブジェクト11であるか被ファクトリオブジェクト12、13であるかの判断が行われる。

【0049】オブジェクトが被ファクトリオブジェクト

12、13である場合は、被ファクトリオブジェクト12、13を生成したファクトリオブジェクト11のオブジェクト名を入手する（ステップS13）。そして、入手したファクトリオブジェクト11のオブジェクト名と被ファクトリオブジェクト12、13に与えられた名前とを連結したオブジェクト名を生成する（ステップS14）。

【0050】図1では、ファクトリオブジェクト11には/factory0というオブジェクト名が既に与えられている。従って、被ファクトリオブジェクト12に対してobject0というオブジェクト名を与えると、名前作成装置10は、/factory0/object0という階層的なオブジェクト名を作成する。但し、オブジェクト名は論理的に階層的な名前空間を与える名前ならば任意である。ここに示した/（スラッシュ）による階層的な名前表現は、論理的な階層性にのみ意味があり、/（スラッシュ）による階層表現に限定されるものではない。

【0051】アプリケーション5は、以上のように生成された被ファクトリオブジェクト12、13のオブジェクト名と、作成された被ファクトリオブジェクト12、13のオブジェクト名と、をネームクライアント14に渡し、ネームサーバ4のネームデータベース16に登録するよう要求する。ネームクライアント14は、このオブジェクト名とオブジェクトリファレンスをネームデータベース管理装置8に送信し、ネームデータベース管理装置8がこれらをネームサーバ4のネームデータベース16に登録する。ネームデータベース16に登録される組は、オブジェクト名と、オブジェクトリファレンス（パーシステントオブジェクトリファレンス21又はトランジエントオブジェクトリファレンス22）と、登録時刻とのフィールドにより構成される。

【0052】ネームクライアント14からオブジェクト名とオブジェクトリファレンスの登録の依頼を受けると、ネームサーバ4のネームデータベース管理装置8が、オブジェクト名とオブジェクトリファレンスに、オブジェクト名とオブジェクトリファレンスを受け取った時刻を登録時刻として付加した組（登録情報）を、ネームデータベース16に登録する。但し、既に同じオブジェクト名を持つ組が存在した場合、ネームデータベース管理装置8は、ネームクライアント7に重複エラーを返すか、アプリケーション5が組の強制登録を要求している場合は、既に同じオブジェクト名を持つ古い組を削除して、強制的に新しい組を登録する。

【0053】図5は、ネームデータベース管理装置8が、ネームデータベース16に既に存在する古い組を削除する時の動作を示すフローチャートである。このフローチャートに従って動作を説明する。まず、ネームデータベース16中に存在する古い組がファクトリオブジェクト11であるか、被ファクトリオブジェクト12、13であるかを判断する（ステップS21）。この判断

は、オブジェクトリファレンス中に生成時刻に関するフィールドがもたれているかどうかで判断する。

【0054】次に、組がファクトリオブジェクト11に対するものであった場合、ファクトリオブジェクト11によって生成された被ファクトリオブジェクト12、13の検索を行ない(ステップS22)、被ファクトリオブジェクト12、13があるか否かを確認し(ステップS23)、被ファクトリオブジェクト12、13がない場合は、ファクトリオブジェクト11のみを削除して(ステップS25)処理を終了する。一方、被ファクトリオブジェクト12、13があった場合は、被ファクトリオブジェクト12、13を削除し(ステップS24)、全ての被ファクトリオブジェクト12、13が削除されると、さらにファクトリオブジェクト11を削除して(ステップS25)、処理を終了する。

【0055】ここで、被ファクトリオブジェクト12、13の検索は、オブジェクト名によって行なう。すなわち、被ファクトリオブジェクト12、13のオブジェクト名はそれを生成したファクトリオブジェクト11のオブジェクト名を連結したものである、という事実から被ファクトリオブジェクト12、13を特定することができる。

【0056】図1では、ファクトリオブジェクト11のオブジェクト名は/factory0である。従って、この組を削除する場合、/factory0をオブジェクト名の最初の要素を持つ組を検索する。これによって、/factory0/object0をオブジェクト名を持つオブジェクトが、ファクトリオブジェクト11によって生成された被ファクトリオブジェクト12であることが判る。本発明によるネームデータベース管理装置8は、ファクトリオブジェクト11とそれによって生成された被ファクトリオブジェクト12の組を一度に削除する。図1では、ネームデータベース16内の/factory0をオブジェクト名を持つ組を削除すると/factory0/object0も同時に削除される。

【0057】以上のように本実施の形態によれば、アプリケーション5の終了時に、ネームサーバ4のネームデータベース16に登録してある組を削除する際、ファクトリオブジェクト11の組を削除することにより、ファクトリオブジェクト11が生成した被ファクトリオブジェクト12、13の組も同時に削除することができる。従って、被ファクトリオブジェクト12、13の組を削除するための要求を省略することが可能であり、アプリケーション5の終了時間を削減できる。また、ファクトリオブジェクト11及び被ファクトリオブジェクト12、13の組を削除する時に、ネットワークを通じてファクトリオブジェクト11の組を削除する要求だけすれば良く、ネットワークトラフィックを削減することができる。

【0058】さらに、アプリケーション5が異常終了し、先の実行で登録された組を正常に削除できなかつた

場合には、アプリケーション5の再度起動の際に先の実行で登録された組を削除する必要があるが、この場合も、同様にしてファクトリオブジェクト11及び被ファクトリオブジェクト12、13を同時に削除できることから、短い時間でアプリケーション5を再起動することができるとともに、ネットワークトラフィックを削減することができる。

【0059】実施の形態2、図6は、実施の形態2のネームサーバ30の構成図を示したものである。ネームサーバ30以外の構成については、図1と同様であることから、ここでは図示を省略した。ここで、実施の形態1と大きく異なる点は、ネームサーバ30が管理するネームデータベース40に同一オブジェクトに対応する組を重複して存在させることができる点にある。図6では、ファクトリオブジェクト名/factory0と被ファクトリオブジェクト名/factory0/object0のそれぞれに対して2つの組が同時に存在する。ネームデータベース40には、図1のネームデータベース16の組の構成に加えて、削除対象の組であるか否かを表すフィールドが存在する。

【0060】低優先度スレッド60は、ネームデータベース40内の必要のなくなった組を削除するスレッドであり、削除対象としてマークされた組(図6中で削除とマークされた組)を削除する。低優先度スレッド60は、これらのスレッドが提供する優先度機構によって、低い優先度で動作するスレッドになっている。すなわち、低優先度スレッド60はコンピュータ1のCPUが他の処理をしていないアイドリング状態にある時に動作が行われる。

【0061】次に、アプリケーション5の起動が始まってからの動作について説明する。アプリケーション5が起動した際、ファクトリオブジェクト11が生成され、ファクトリオブジェクト11に対応するオブジェクト名とオブジェクトリファレンスがネームサーバ30のネームデータベース40に登録される。

【0062】次に、アプリケーションクライアント6が、ファクトリオブジェクト11に対して別のオブジェクトを生成するように要求を送信する。この要求に応じて、ファクトリオブジェクト11によって被ファクトリオブジェクト12が生成される。さらに、被ファクトリオブジェクト12のオブジェクトリファレンスが生成され、オブジェクト名が作成されてファクトリオブジェクト11と同様にしてネームサーバ30のネームデータベース10にオブジェクトリファレンスとオブジェクト名が登録される。ここまででは、実施の形態1の動作と同一である。

【0063】アプリケーション5が終了する際、本実施の形態におけるネームサーバ30のネームデータベース40に対しては、組の削除を行なわない。すなわち、アプリケーション5が正常に終了する際、組の削除の要求

をネームサーバ30に対して送ることなく終了する。次に、アプリケーション5を再起動した際、新しい組がネームデータベース40に追加される。この時、ネームデータベース管理装置50が、同一オブジェクト名を持つ既存の古い組の存在を検出した場合、それらは削除対象としてマークされ、削除用のスレッドである低優先度スレッド60が起動される。

【0064】例えば、オブジェクト名/factory0の新しい組を登録する場合、時刻13:46:35.98に登録されている古い組が既に存在する。この場合、時刻13:46:35.98に登録された古い組は削除対象としてマークされ、低優先度スレッド60が起動される。そして、新しい組がネームデータベース40に追加される。図6では、新しい組は時刻14:02:03.44に登録されている。

【0065】ネームデータベース管理装置50によって起動された低優先度スレッド60は、古いファクトリオブジェクト11の組を削除し、さらに古いファクトリオブジェクト11によって生成された被ファクトリオブジェクト12の組を検索し、これも同様に削除する。図7は、低優先度スレッド60が古い被ファクトリオブジェクト12の組を検出し、削除する時の動作手順を示すフローチャートである。このフローチャートに従って動作を説明する。

【0066】まず、低優先度スレッド60は、古いファクトリオブジェクト(旧ファクトリオブジェクト)11に対応する組の削除要求があるかチェックし(ステップS31)、古い被ファクトリオブジェクト(旧被ファクトリオブジェクト)12に対応する組の削除要求があった場合には、新しいファクトリオブジェクトに対応する組の登録時刻をネームデータベース40より入手する(ステップS32)。次にオブジェクト名から被ファクトリオブジェクトに対応する組を検索する(ステップS33)。

【0067】古い被ファクトリオブジェクト12に対応する組がある場合は(ステップS34)、この組のオブジェクトリファレンス内にあるオブジェクトの生成時刻(トランジエントオブジェクトリファレンス22内の時刻)を入手し(ステップS35)、先に獲得した新しいファクトリオブジェクトに対応する組の登録時刻と比較を行う(ステップS36)。オブジェクトリファレンス内の時刻の方が、先に獲得した新規ファクトリオブジェクトの登録時刻よりも古い場合、このオブジェクトを古いファクトリオブジェクト11によって生成された被ファクトリオブジェクト112に対応する組であると判断し、これを削除する(ステップS37)。

【0068】例えば、図6では、新しいファクトリオブジェクトの組がオブジェクト名/factory0で登録される際、/factory0をオブジェクト名の最初のコンポメントとして持つ被ファクトリオブジェクトを検索する。例えば、時刻13:47:10.22に登録されたオブジェクト名/fac

ory0/object0の組がそれである。そして、このオブジェクトリファレンス内の時刻と、新しいファクトリオブジェクトの組の時刻14:02:03.44を比較する。オブジェクトリファレンス内の時刻の方が古い場合、この被ファクトリオブジェクトの組が削除の対象と判断され削除される。

【0069】以上のように本実施の形態によれば、アプリケーション5は、正常終了、異常終了のいずれの終了でもネームサーバ30に組の削除の要求を出す必要がなく、ネットワークトラフィックの削減を行うことが可能である。また、アプリケーション5が異常終了した場合、既存の古い組がネームデータベース40内に残るが、新しい組と古い組の両方がネームデータベース40内に存在できるため、古い組の削除を待つことなく、新しい組をネームデータベース40に登録することが可能である。これによって、異常終了した際のアプリケーション5の回復時間を大幅に短縮することが可能である。また、アプリケーション5では、ネームサーバ30のネームデータベース40から既存の古い組を削除するための命令コードを記述する必要がなく、従来よりも、少ないコードでアプリケーション5を記述できる。

【0070】実施の形態3. 図8は、ネームデータベースキャッシュ70が存在したネーミングサービスの構成を示した図である。図において、アプリケーション5が動作するコンピュータ2については、図1と同様のため、ここでは図示を省略した。

【0071】ネームデータベースキャッシュ70は、ネームサーバ4のネームデータベース16中のオブジェクト名とオブジェクトリファレンスをコピーしたものである。アプリケーションクライアント6はオブジェクトリファレンスを入手する際に、オブジェクト名をネームクライアント7を通じてネームサーバ4に渡す。ネームサーバ4のネームデータベース管理装置8は、与えられたオブジェクト名をもとに該当するネームデータベース16中のオブジェクトリファレンスを検索し、この検索したオブジェクトリファレンスをネームクライアント7を通じてアプリケーションクライアント6に返す。

【0072】ネームクライアント107は、返されたオブジェクトリファレンスに、このオブジェクト名とオブジェクトリファレンスを入手した時刻を登録時刻として加えたキャッシュ保存情報を、将来の再利用のためネームデータベースキャッシュ70に保存する。

【0073】ネームデータベースキャッシュ70は、無効なオブジェクト名とオブジェクトリファレンスを持つ場合がある。例えば、アプリケーション5が再起動されたとする。/factory0をオブジェクト名に持つファクトリオブジェクト11に対応するオブジェクト名とオブジェクトリファレンスは無効ではないかもしれないが、/factory0/object0をオブジェクト名に持つ被ファクトリオブジェクト112に対応するオブジェクト名とオブジェ

クトリファレンスは既に無効である。なぜなら、再起動によって被ファクトリオブジェクト12は消滅しているからである。

【0074】ネームデータベース管理装置8は、ネームデータベース16に登録されているファクトリオブジェクト11に対応する組に変更があった時、定期的又はユーザの要求によっては即座に（いずれでも良い）、この組の変更に基づいてオブジェクト名とオブジェクトリファレンスによる更新情報をネームクライアント7に伝える。伝える方法は、マルチキャストでもユニキャストでも方法は問わない。

【0075】一方、被ファクトリオブジェクト12に対応する組に変更があつても、それは伝えない。例えば、アプリケーション5が再起動された場合、オブジェクト名/factory0のファクトリオブジェクト11の組がネームデータベース16に新たに登録された場合、新たに登録された組のオブジェクト名とオブジェクトリファレンスによる更新情報をネームクライアント7に伝えるが、再起動によって、オブジェクト名/factory0/object0の被ファクトリオブジェクト12の組が消滅したことについては、ネームクライアント7には伝達しない。

【0076】図9は、ネームデータベース16に登録されている組に変更（更新）があつた時に、ネームデータベース管理装置8がネームクライアント7に更新情報を伝達する時の動作手順を示すフローチャートである。このフローチャートに従って動作を説明する。まず、更新があつたかどうかをチェックする（ステップS41）。次に、更新があつた場合には、この更新がファクトリオブジェクト11に対するものかどうかをチェックする（ステップS42）。このチェックは、オブジェクトリファレンス中に生成時刻情報があるかどうか（トランジエントオブジェクトリファレンス22かどうか）を調べることで可能である。

【0077】ここで、オブジェクトリファレンスがパーシステントオブジェクトリファレンス21であり、ファクトリオブジェクト11であることが判明した場合、ファクトリオブジェクト11の更新情報をオブジェクト名とオブジェクトリファレンスをネームクライアント7に伝える（ステップS43）。ファクトリオブジェクト11でなかった場合は、再び別の更新があつたかどうかをチェックする。

【0078】ネームクライアント7は、ネームサーバ4のネームデータベース管理装置8からファクトリオブジェクト11に対応する組のオブジェクト名とオブジェクトリファレンスの更新情報を受け、この変更情報を入手した時刻を登録時刻として加えた新規キャッシュ保存情報を、ネームデータベースキャッシュ70に既に保存されている新規キャッシュ保存情報を該当するオブジェクト名とオブジェクトリファレンスと登録時刻によるキャッシュ保存情報を上書きする。

【0079】また、ファクトリオブジェクト11のオブジェクト名から、既に保存されているファクトリオブジェクト11に対応する被ファクトリオブジェクト12のキャッシュ保存情報を検索し、これらを全て削除する。例えば、オブジェクト名/factory0のファクトリオブジェクト11に対するオブジェクト名とオブジェクトリファレンスの更新情報がネームサーバ4から送られた場合、この送られた更新情報に基づいた新規キャッシュ保存情報を、ネームデータベースキャッシュ70に既に保存されている同一オブジェクト名を持つキャッシュ保存情報に上書きする。

【0080】次に、オブジェクト名/factory0を用いて、ファクトリオブジェクト11によって生成された被ファクトリオブジェクト12のキャッシュ保存情報を検索する。図8のネームデータベースキャッシュ70では、オブジェクト名/factory0/object0がこの検索されたものである。そして、検索されたキャッシュ保存情報を全てネームデータベースキャッシュ70から削除する。この動作を以下で説明する。

【0081】図10は、上記のネームクライアント7がネームデータベースキャッシュ70に新規キャッシュ保存情報を上書き、又は既存のキャッシュ情報を削除する時の動作手順を示すフローチャートである。このフローチャートに従って動作を説明する。

【0082】ネームサーバ4からの更新通知があつた場合（ステップS51）、ネームデータベースキャッシュ70に既に保存されているファクトリオブジェクト11のオブジェクト名とオブジェクトリファレンスと登録時刻によるキャッシュ保存情報を、新たなオブジェクト名とオブジェクトリファレンスと登録時刻による新規キャッシュ保存情報を上書きする（ステップS52）。ファクトリオブジェクト11のオブジェクト名から被ファクトリオブジェクト12に対応するキャッシュ保存情報をネームデータベースキャッシュ70より検索する（ステップS53）。被ファクトリオブジェクト12のキャッシュ保存情報があつた場合は、このキャッシュ保存情報を削除する（ステップS54～S55）。

【0083】以上のように本実施の形態によれば、ファクトリオブジェクト11のみの更新情報を伝播し、この更新情報に基づく新規キャッシュ保存情報をネームデータベースキャッシュ70に保存されているキャッシュ保存情報を上書きしており、ネームデータベースキャッシュ70に登録されている被ファクトリオブジェクト12のキャッシュ保存情報については、更新情報中のファクトリオブジェクト11のオブジェクト名からネームデータベースキャッシュ70内を検索しローカルに削除ができる。これによって、ネームデータベースキャッシュ70に保存されているキャッシュ保存情報を変更する時のネットワークトラフィックを大幅に削減することが可能になる。

【0084】また、ファクトリオブジェクト11の更新通知が来た時点で、ファクトリオブジェクト11によって生成された被ファクトリオブジェクト12の無効なキャッシュ保存情報を認識することができる。これによって、早期に、ネームデータベースキャッシュ70内のキャッシュ保存情報の無効性を発見することができ、アプリケーション5に対する無駄なアクセスを削減することができる。

【0085】実施の形態4、実施の形態3では、ネームサーバ4からのファクトリオブジェクト11に対応する更新情報には、オブジェクト名とこのオブジェクト名に対応するオブジェクトリファレンスが含まれていた。しかし、本実施の形態では、ファクトリオブジェクト11に対するオブジェクトリファレンスはパーシステントオブジェクトリファレンス21である（再起動行っても不变）という事実を利用して、オブジェクト名だけを更新情報に入れてネームクライアント7に更新通知を伝えるものである。これによって、ネームデータベースキャッシュ70中のファクトリオブジェクト11のオブジェクト名の上書き、被ファクトリオブジェクト12のオブジェクト名の削除ができる、ネームデータベースキャッシュ70の更新によるネットワークトラフィックをさらに削減できる。

#### 【0086】

【発明の効果】この発明は、以上説明したように構成されているので、以下に示すような効果を奏する。

【0087】第1の発明では、ファクトリオブジェクトと被ファクトリオブジェクトのそれぞれに対して生成された識別情報と、それぞれに対して作成された名前とをネームデータベースに登録し、登録された識別情報と名前の削除時に、ファクトリオブジェクトの識別情報と名前を削除し、同時に被ファクトリオブジェクトの識別情報と名前を削除するので、ネットワークを通じてファクトリオブジェクトの識別情報と名前の削除要求をするだけで被ファクトリオブジェクトの識別情報と名前を同時に削除することができ、ネットワークトラフィックの削減がかかる。

【0088】第2の発明では、被ファクトリオブジェクトの識別情報に被ファクトリオブジェクトの生成された生成時刻が付加され、この生成時刻に基づいて上記ネームデータベースに登録されている被ファクトリオブジェクトの有無を確認し、確認された被ファクトリオブジェクトの識別情報と名前を削除するので、ネームサーバが単独で削除対象の被ファクトリオブジェクトの識別情報と名前を削除することができるので、ネットワークを通じて被ファクトリオブジェクトの識別情報と名前の削除

要求をする必要がなくネットワークトラフィックの削減がかかる。

【0089】第3の発明では、ネームデータベースに登録されたファクトリオブジェクトの識別情報と名前に更新があった時に、この更新されたファクトリオブジェクトの新たな識別情報と名前を、ネームデータベースキャッシュに送信し、ネームデータベースキャッシュに既に保存されている上記ファクトリオブジェクトの識別情報と名前に上書きし、上記ネームデータベースキャッシュに既に保存されている上記ファクトリオブジェクトに対応する上記被ファクトリオブジェクトの識別情報と名前を削除するので、被ファクトリオブジェクトの識別情報と名前の削除要求をする必要がないので、ネームデータベースキャッシュ更新のためのネットワークトラフィックの削減がかかる。

#### 【図面の簡単な説明】

【図1】 実施の形態1のネーミングサービスの構成図。

【図2】 実施の形態1におけるオブジェクトリファレンスの構成図。

【図3】 実施の形態1におけるオブジェクトリファレンス生成装置の動作を示すフローチャート。

【図4】 実施の形態1における名前作成装置の動作を示すフローチャート。

【図5】 実施の形態1におけるネームデータベース管理装置の動作を示すフローチャート。

【図6】 実施の形態2におけるネームサーバの構成図。

【図7】 実施の形態2におけるネームデータベース管理装置の動作を示すフローチャート。

【図8】 実施の形態3のネーミングサービスの構成図。

【図9】 実施の形態3におけるネームデータベース管理装置の動作を示すフローチャート。

【図10】 実施の形態3におけるネームクライアントの動作を示すフローチャート。

【図11】 従来のネーミングサービスの構成図。

【図12】 従来のネーミングサービスの動作説明図。

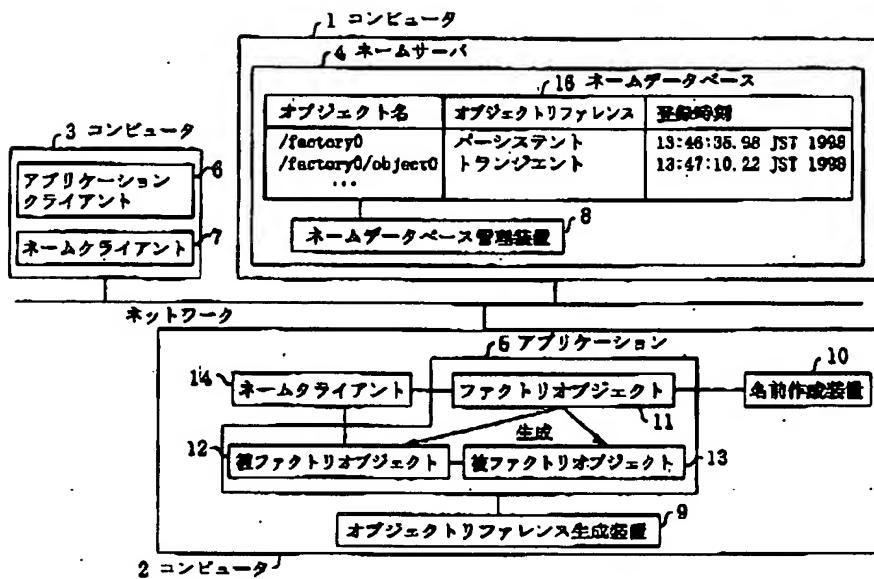
#### 【符号の説明】

- 1、2、3 コンピュータ、4 ネームサーバ、5 アプリケーション、6 アプリケーションクライアント、7 ネームクライアント、8 ネームデータベース管理装置、9 オブジェクトリファレンス生成装置、10 名前作成装置、11 ファクトリオブジェクト、12、13 被ファクトリオブジェクト、14 ネームクライアント。

(12)

特開2000-215179

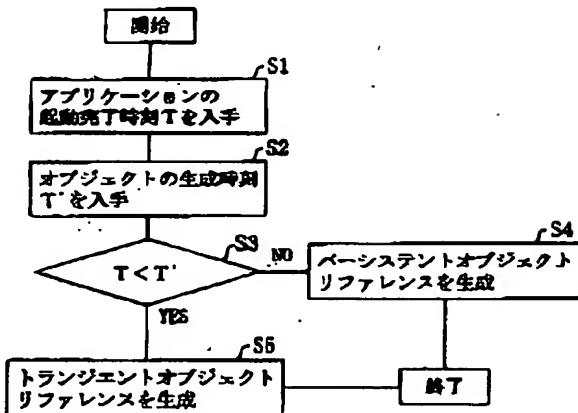
【図1】



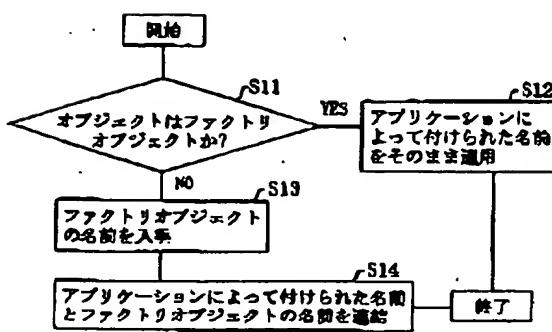
【図2】

21 ベーシステントオブジェクトリファレンス
ホスト名 ポート番号 インタフェースなど
ホスト名 ポート番号 時刻
22 トランジエントオブジェクトリファレンス

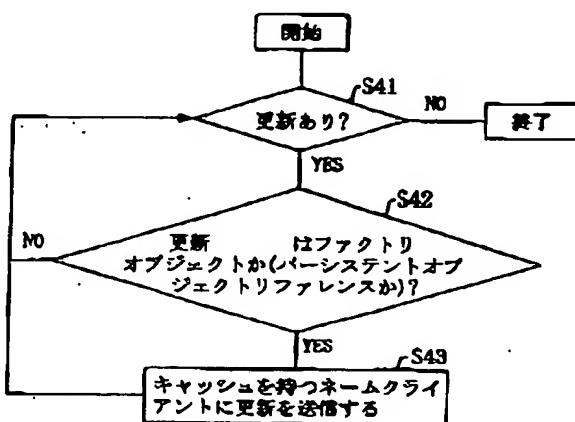
【図3】



【図4】



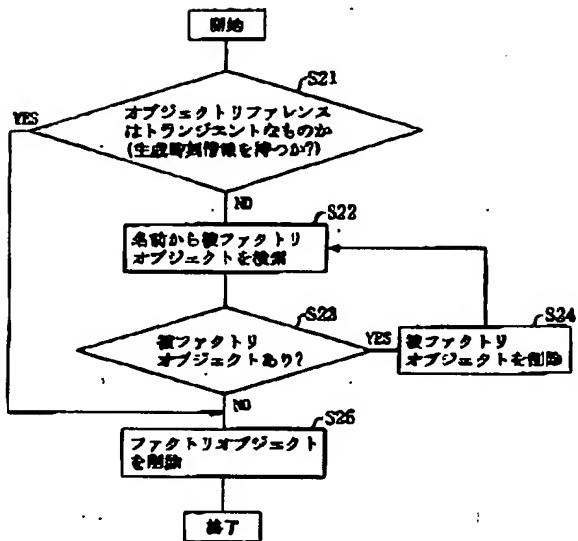
【図9】



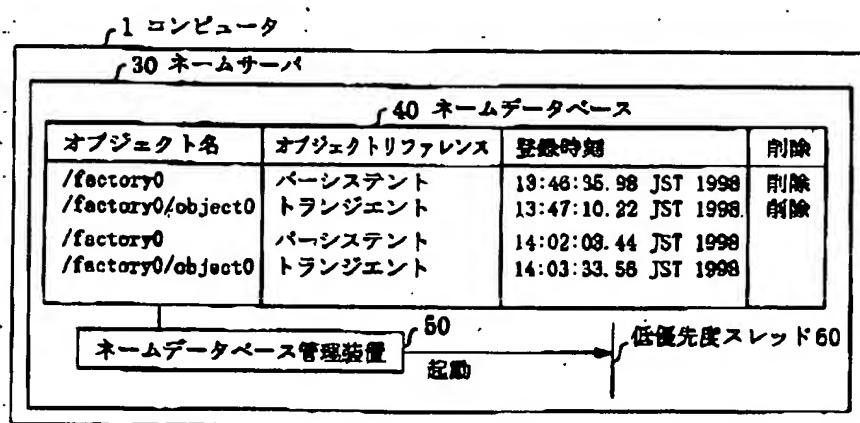
(13)

特開2000-215179

【図5】



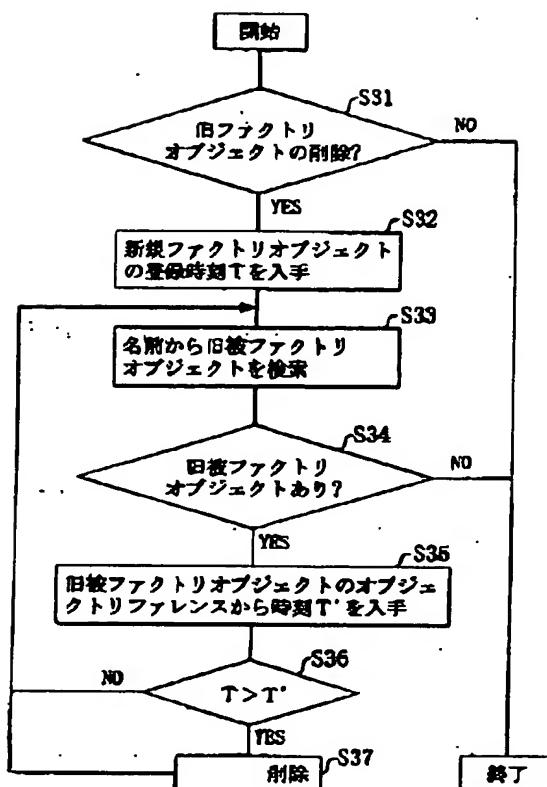
【図6】



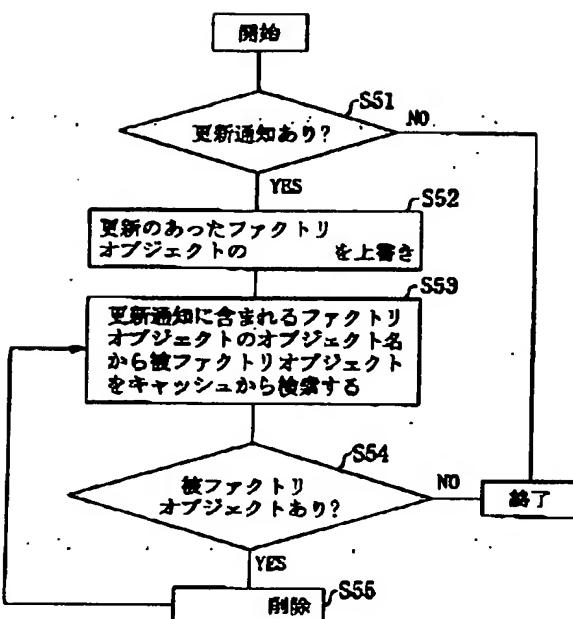
(14)

特開2000-215179

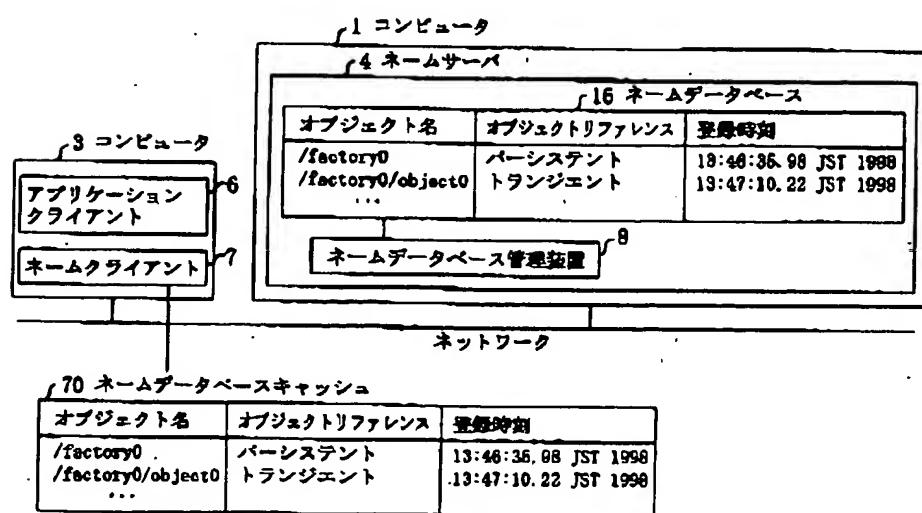
【図7】



【図10】



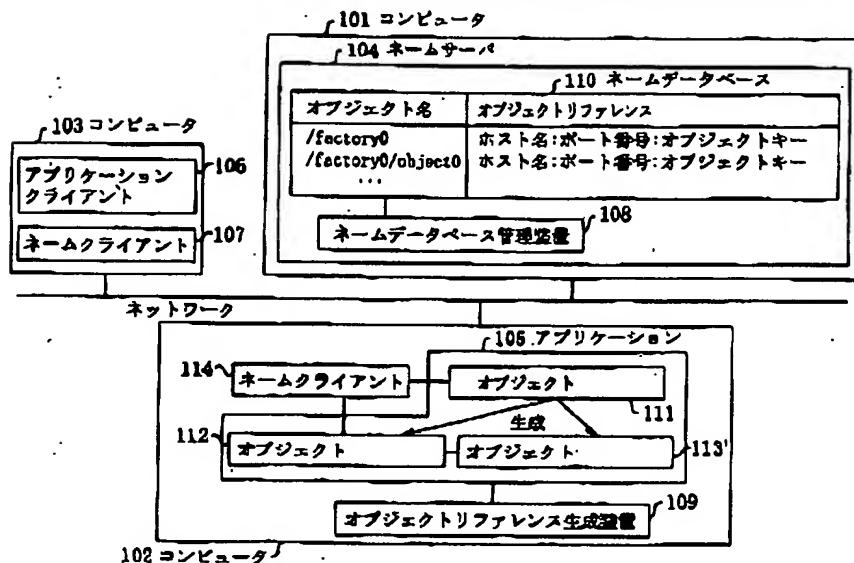
【図8】



(15)

特開2000-215179

【図11】



【図12】

